Lab 12: Analysis of Chrome browsing history with Python

In this short final lab, you will apply and extend the ideas presented in the lecture in order to query several SQLite databases using Python.

You are provided with the Chrome user data from a forensic image. Follow the suggestions below to find out about some of the user's activities.

1. The (imaginary) context

In an investigation into a number of car thefts in and around Edinburgh, Police Scotland have identified several suspects. The police think that they are working together.

The laptops of all three suspects were seized and forensic images made. You have been provided with the file **Chrome.zip** (moodle). This contains the Google Chrome directory for one of the suspects complete with all his Chrome user data.

2. Exploring the history database (structure)

The first task is to view this database in a suitable database viewer so you can see how it is organised.

Download DB Browser for SQLite from http://sqlitebrowser.org/ - there are several versions available, windows executables for 32 and 64 bit and also a portable app which requires no installation. Choose whichever is best for you.

Unzip Chrome.zip.

In DB Browser, open the file **History** in the **Chrome\User Data\Default** directory.

a) Which tables does this file contain?

b) Look at the **downloads** table. What does it contain?

c) Now inspect in detail the **urls** table. What data does it contain and how does it store this exactly?

Record information about every column a table in a table similar to the one shown overleaf. Refer to https://www.lowmanio.co.uk/blog/entries/how-google-chrome-stores-web-history/ for further information.

Column name	explanation	Example data
id	Unique identifier for each url	3
url	the url visited	https://www.google.co.uk/search?q=how+to+install+cortana&oq=how+to+install+cortana&aqs=chrome..69i57j0l5.3875j0j8&sourceid=chrome&ie=UTF-8
…	…	…
favicon_id	…	…

d) Are the urls and visits tables related? How exactly are they linked?

3. Exploring the history database (contents)

Now look at the contents of the tables. To answer the questions below, you may also need to look in tables that you have not previously explored, such as the table keyword_search_terms. Use the filters at the top of each column to help with these searches.

a) To which directory were the downloads saved? Does this suggest the name of the user whose history this is?

b) What is the Spotify user name?

c) What is the name of the Outlook mail account accessed by the user? (look in the title column, urls table)

d) The urls table also shows that a Gmail inbox was accessed. What is the gmail address of this Inbox? When was it last accessed?

e) Does the browsing history confirm that the user is a car enthusiast? Give some examples.

f) Does the browsing history support the police's suspicion that the user is involved in illegal activities? In organised crime? Give 3 example database entries that support this suspicion.

g) Is there any evidence that suggests the user may be planning to leave the country?

4. Summarising browsing history using Python: sites visited

Karthikeyan KC posted code for analysing Chrome browsing history at https://geekswipe.net/technology/computing/analyze-chromes-browsing-history-with-python/.

Read the post and make sure you understand it.

Make a copy of the code and adapt it to graph an overview of the sites visited by the suspect.

a) Which two tables from the history file are used by the code? Why?

b) Does your graph look as neat as the example shown on the website?

c) Do you need to limit the sites included?

5. Summarising browsing history using Python: activity timeline

Now that you have some experience using Python to work with the urls and visits table, it's time to apply this.

Write a script to count the number of sites visited every day, and output it like the table below:

date	number of webpages visited
31/1/2017	5
…..	
31/3/2017	56

(the first and last date are correct, but the counts are not)

For this, you only need the visits table.

The main challenges here are:

- Writing the SQL query needed to extract the relevant information from the table.
  Hint: Refer to the code you used in the previous exercise and to the example SQL queries shown in the lecture for querying satnav data.
- Decoding the timestamps.
  Hint: Refer to the lecture for how to do this - you can either do it in the SQL you use to extract the data from the table, or you can do it afterwards using Python itself.
- Summarising the information - adding up the number of pages visited.
  Hint: From each timestamp, use just the date (not the time). Create a dictionary with the dates as keys and the number of pages as value. If you set this as 0 initially, you can then add 1 to the value for the relevant date as you step through each of the entries in the visits table.

Optional extension: create a chart (e.g. bar chart) to show this information visually.